

REMARKS

This is intended as a full and complete response to the Office Action dated October 31, 2008, having a shortened statutory period for response set to expire on January 31, 2009. Please reconsider the claims pending in the application for reasons discussed below.

Claims 10-20, 33-42, 45-47, 50 and 51 are pending in the application. Claims 10-20, 33-42, 45-47, 50 and 51 remain pending following entry of this response.

Claim Rejections - 35 U.S.C. § 112

The Examiner rejects claims 10 and 33 under 35 § U.S.C. 112, first paragraph, as failing to comply with the enablement requirement as follows:

[claims 10 and 33 contain] subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 10 recite the limitation "receiving, at a server, a first request from a client, wherein the first request is a request to invoke a remote procedure call at the server; receiving, at the server, a second request from the client, wherein the second request comprise an internationalization context for processing the first request." The specification fails to describe the invention in such terms that one skilled in the art can make and use the claimed invention. The information contained in the disclosure of the application is not sufficient to inform those skilled in the relevant art the nature and the cause of the second request.

Office Action, p. 2. Respectfully, Applicants traverse this rejection. At paragraph 9, the specification provides:

a server computer receives a first request from a remote client computer. Illustratively, the server computer then receives a second request from the client computer wherein the second request comprises an internationalization context comprising the client's preferred conventions for processing the first request. The server computer associates the second request with the first request for every thread of processing.

Further, Figure 5 provides an example of this claimed process. The description Figure 5 provides as follows:

Figure 5 illustrates the propagation of the internationalization context 506 from a client request to a remote server. The client 502 is located in a French locale and CET (central European time) time zone. The internationalization context 506, comprising the client's locale and time-zone information, is transmitted separately from the called method function m1(...). Upon receiving the client request at the server 510 in the Japan locale and JST (Japanese standard time) time zone, the server 510 will extract the client's internationalization context and process the request using the client's internationalization context. In this example, it is necessary to further process the client request at other remote servers 512, 514 located in different locales and time zones. As illustrated, the client's internationalization context 506 propagates with each successive call to a remote server. With each successive call to a remote server, each remote server will extract the locale specific information in the client's internationalization context 506 and process the request using the information.

Specification, ¶ 0039. Further still, Figure 6A illustrates an embodiment of the internationalization context data structure described as being “transmitted separately from the called method function m1(...).” Clearly these passages specify “the nature of the second request” as being an “internationalization context” that is “transmitted separately from the called method function m1(...),” i.e., transmitted separately from a requires to invoke the remote procedure call “m1(...).” Further still, Figures 7 and 8 illustrate methods for a client to generate and send the first and second requests request and for a server to receive and process the first and second requests (and propagate the request to additional servers). Furthermore, paragraphs 0041-0046 provide example implementations of the claimed invention using particular standards e.g., using CORBA (common object request broker architecture) which may be implemented using IBM's SOM (system object model) and DSOM (distributed system object model), as well as an example implemented using JAVA technology developed by Sun Microsystems.

For all the foregoing reasons, Applicants submit that the specification includes ample material to satisfy the requirements of 35 § U.S.C. 112, first paragraph relative to claims 10 and 33. Accordingly, Applicants respectfully request that this rejection be withdrawn.

Claim Rejections - 35 U.S.C. § 103

Claims 10-14, 17, 33-37, 40, 45, 47, 50-51 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application No. 2002/0162093 to *Zhou et al*, in view of U.S. Patent No. 6,430,607 to *Kavner* and further in view of U.S. patent Application No. 5,404,523 to *DellaFera et al*.

The Examiner bears the initial burden of establishing a prima facie case of obviousness. See MPEP § 2141. Establishing a prima facie case of obviousness begins with first resolving the factual inquiries of *Graham v. John Deere Co.* 383 U.S. 1 (1966). The factual inquiries include:

- (A) determining the scope and content of the prior art;
- (B) ascertaining the differences between the claimed invention and the prior art;
- (C) resolving the level of ordinary skill in the art; and
- (D) considering any objective indicia of nonobviousness.

Once the *Graham* factual inquiries are resolved, the Examiner must determine whether the claimed invention would have been obvious to one of ordinary skill in the art.

Respectfully, Applicant submits that the Examiner has not properly characterized the teachings of the references and/or the claims at issue. Accordingly, a prima facie case of obviousness has not been established.

In particular, Applicants submit that *Zhou*, *Kavner* and *Dellafera* do not teach, show, or suggest the method recited by claim 10: “operative in a distributed computing environment having clients and a plurality of servers located across geographically dispersed boundaries” that includes:

receiving, at a server, a first request from a client, wherein the first request is a request to invoke a remote procedure call at the server;
receiving, at the server, a second request from the client, wherein the second request comprises an internationalization context for processing the first request, wherein the internationalization context specifies geographically specific parameters set for the client;
propagating the first request with the attached internationalization context from the server to an application associated with an application interface on a second server.

Independent claims 33 and 45 recite similar limitations. The Examiner concedes that these limitations are not taught by *Zhou* or *Dellafera*, but instead suggests:

Kavner teaches if the remote request requires additional static parameters, the client application returns to state 1310 and again calls the AddParam routine (See col. 3, lines 42-55, col. 24, lines 56-59, col. 33, lines 27-57, *receiving, at a server a second request comprising additional parameters*).

It would have been obvious to one with ordinary skill in the art at the time the invention was made to incorporate the teaching of *Kavner* in the claimed invention of *Zhou* in order to allow the use of incremental data blocks (See col. 3, lines 42-54).

See Office Action, p. 5. The passage at *Kavner* 3:42-55 indicates that a client may “begin using incremental data blocks” before receiving a “entire data blocks,” and provides a supporting example of an image being downloaded to a client. That is, a client (e.g., a web browser) might begin displaying an image before receiving the complete image. This passage also points out that conventional RPC mechanisms require that a server provide a complete response to a remote procedure call (e.g., transmit a complete image) before the client may process any data associated the result, i.e., the entire image (or other “static data”) must be received prior to displaying any portion of the image (or otherwise process the “static data”). Further, the cited passage points out that this process leads to greater latency in application responsiveness and wastes memory resources.

At the same time, the passage clearly describes aspects of a single remote procedure call; namely, that “the client waits until receiving the entire image before attempting to display the image,” *Kavner*, 3:46-48, and also describes a couple flaws with this approach (latency and wasteful use of memory). Thus, nothing in the this description of the shortcomings of conventional RPC facilities discloses the claimed steps of receiving both a first and second request, as characterized by the present claims.

The remaining passages from *Kavner* cited by the Examiner go on to describe an improvement for an RPC system where a remote procedure may be generated that includes the use of “static parameters.” Specifically, *Kavner* teaches that:

remote requests are managed by a request layer. In the preferred embodiment, the request layer 206 is called the Microsoft Network Procedure Call and is referred to herein as the "MPC layer 206." The MPC layer 206 also includes a client application programming interface ("the client MPC layer 206a"), and a service application programming interface ("the service MPC layer 206b and 206c") for interfacing with client and service applications. The client MPC layer 206a and the service MPC layers 206b and 206c contain a variety of routines which allow the client and service applications to communicate with each other.

Kavner, 8:35-46. And further,

the names of the routines 900 existing in the client MPC layer 206a that implement the novel features of the present invention include: CreateServiceInstance, CreateMethodInstance, AddParam, RequestDynamicParam, Execute, CancelExecution, WaitIncremental, GetBuffer, and FreeMemory. Each of these routines is further discussed below.

Kavner, 17:53-59. More simply, *Kavner* provides a remote procedure call facility where an "MPC layer" is used to generate remote procedure calls to a given service provided by a server. *Kavner* includes an example of calls to a "CHAT" service (allowing users to communicate with one another) and a "WEATHER" service (allowing users to download a weather map). As described, when a user requests a weather map, the MPC layer is used to add a collection of parameters to an RPC request ultimately sent to a remote server. *Kavner* teaches that the self-styled "AddParam" routine may be used to submit any number of static parameters to the RPC layer.

Proceeding to state 1312, if the remote request requires additional static parameters, the client application returns to state 1310 and again calls the AddParam routine. For example, when the client WEATHER application 204a requests the "download weather map" method, the client application 200a passes a variety of static parameters that identify the requested weather map.

Kavner, 24:57-63. In other words, the client submits parameters to the MPC layer (on the client), and in response, the MPC layer (also on the client) generates the appropriate RPC request ultimately transmitted to an application server. As the "AddParam" call is used by the client to submit parameters for an RPC call generated by the MPC layer on the client, Applications submit that this RPC facility disclosed in *Kavner* does not disclose "*receiving, at a server a second request comprising additional*

parameters” as suggested by the Examiner. Plainly, the AddParam call is used by the client to supply all needed parameters for an RPC call (e.g., to identify a particular weather map) to the MPC layer which marshals the parameters in an actual request sent to the server. Thus, Applicants submit that the MPC layer in Kavner does not teach, show, or even suggest, the claimed steps of:

receiving, at a server, a first request from a client, wherein the first request is a request to invoke a remote procedure call at the server;
receiving, at the server, a second request from the client, wherein the second request comprises an internationalization context for processing the first request, wherein the internationalization context specifies geographically specific parameters set for the client; and
propagating the first request with the attached internationalization context from the server to an application associated with an application interface on a second server,

as recited by independent claims 10, 33, and 45

Further still, as recited by claim 10, the step of “processing the first request” includes “providing the first request and internationalization context to an application configured to perform calculations requested by the remote procedure call using the geographically specific parameters included in the internationalization context and further configured to return a result formatted according to a formatting convention selected based on the geographically specific parameters.” The Examiner suggests:

[Zhou teaches] wherein processing the first request comprises: providing the first request and internationalization context to an application configured to perform calculations requested by the remote procedure call using the geographically specific parameters included in the internationalization context and further configured to return a result formatted according to a formatting convention selected based on the geographically specific parameters (See page 1, paragraph [0008], page 3, paragraph [0032-0033], *the framework interacts with the presentation layer to prepare replies for return to the clients in a format and protocol suitable for presentation on the client*);

Office Action, p. 4. However, Zhou, ¶ 33 discusses aspects of a “framework 220” configured to interact with a “business logic layer” and a “presentation layer” to process requests. At no point, however, does this passage teach, show, or suggest, a step where a first request is received from a client (as claimed) or a step where a second request is received from the client, in particular, where the second request includes “an

internationalization context for processing the first request,” as claimed. Set out in full, *Zhou*, ¶ 33 provides as follows:

The framework 220 is composed of a model dispatcher 222 and a request dispatcher 224. The model dispatcher 222 routes client requests to the appropriate business logic in the business logic layer 204. It may include a translator 226 to translate the requests into an appropriate form to be processed by the business logic. For instance, the translator 226 may extract data or other information from the requests and pass in this raw data to the business logic layer 204 for processing. The request dispatcher 224 formulates the replies in a way that can be sent and presented at the client. Notice that the request dispatcher is illustrated as bridging the execution environment 202 and the presentation layer 212 to convey the understanding that, in the described implementation, the execution environment and the presentation layer share in the tasks of structuring replies for return and presentation at the clients.

Zhou, ¶ 33. Plainly, this passage provides a general description of a “framework 220” as including “a model dispatcher 222 and a request dispatcher 224,” and goes on to describe how these elements may generally interact with the “business logic layer 204” and the “presentation layer 212.”

Applicants submit that the general observation by the Examiner that “the framework interacts with the presentation layer to prepare replies for return to the clients in a format and protocol suitable for presentation on the client” fails to demonstrate that the description of the operations of the “framework 220” in fact teach the limitations of the present claims. For example, the general observation that a reply may be prepared “in a format and protocol suitable for presentation on the client,” teach the particular step of “an application ... configured to return a result formatted according to a formatting convention selected based on the geographically specific parameters.” No mention of “selecting a formatting convention selected based on the geographically specific parameters,” is in any way discussed or implied by this passage. At best, this passage describes that the “translator 206 formulates the replies in a way that can be sent and presented at the client.”

Accordingly, for all of the reasons given above, Applicants submit that independent claims 10, 33, and 45 (and the dependent claims) are allowable and respectfully request allowance of the same.

Claims 45, 47, 50-51 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Application No. 2002/0162093 to *Zhou et al* in view of U.S. patent Application No. 5,404,523 to *DellaFera et al*.

Applicants submit that *Zhou* does not disclose a “method for transparently propagating internationalization context information” that includes “receiving, at a first computer, a first request from a second computer, the first request including an internationalization context, wherein the internationalization context specifies geographically specific parameters set for the client computer,” as recited by claim 45.

Zhou discloses a:

compilation and translation system internationalizes an application authored for one locale for use in other locales. The system compiles documents (e.g., web pages, email forms, UI screens, etc.) authored for one locale by automatically extracting locale-sensitive content (e.g., language, regional information, slang, cultural aspects, etc.) into a separate data structure (e.g., a structured text file, database file, etc.). The source code and other locale-independent elements (e.g., formatting data) remain in the compiled document. The extracted content can then be translated for use in other locales. During runtime, requests from different locales can be served locale-sensitive responses by retrieving the compiled document and dynamically populating it with the appropriate content of the target locale.

Zhou, ¶ 8. More simply, *Zhou* discloses that certain documents (e.g., a web page) may be “compiled” by replacing locale specific information with a reference to a “resource bundle.” Paragraphs 0095 and 0096 describe that once such a document is processed, and stored, a web-server, (i.e., the application data manager of *Zhou*, Figure 6) may serve the document to a requesting client. Prior to serving the web-page to the client, however, the references in the “compiled document” are used by the “application data manager to “dynamically populate” the document “with the appropriate content of the target locale.” Nowhere in this process of serving web pages with some portion of dynamic content does *Zhou* disclose:

receiving, at a server, a first request from a client, wherein the first request is a request to invoke a remote procedure call at the server and receiving, at the server, a second request from the client, wherein the second request comprises an internationalization context for processing the first request,

as recited by claim 45. Nevertheless, the Examiner suggests:

As per claim 45, Zhou et al teaches receiving, at a first computer, a first request from a second computer, the first request including an internationalization context, wherein the internationalization context specifies geographically specific parameters set for the client computer (See page 3, paragraph [0033]; extracting the internationalization context from the first request (See page, 6, paragraph [0074]));

Office Action, p. 8. However, as demonstrated above, *Zhou*, ¶ 33 describes how a “framework 220” configured to interact with a “business logic layer” and a “presentation layer” to process requests. At no point, however, does this passage teach, show, or suggest, a step where a first request is received from a client (as claimed) or a step where a second request is received from the client, in particular, where the second request includes “an internationalization context for processing the first request,” as claimed. Accordingly, Applicants submit that these passages do not teach, show, or suggest the specific relationship between a first request and a second request having the specific relationships, as characterized by independent claim 45.

Accordingly, Applicants submit that claims 45, 47, and 50-51 are patentable over *Zhou* in view of *DellaFera*. Applicants respectfully request, therefore, that this rejection be withdrawn.

Claims 15-16, 18-20, 38-39, 41-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent Application No. 2002/0162093 to *Zhou et al* in view of US. Patent No. 6,430607 to *Kavner* and further in view of US. Patent Application No. 5,404523 to *DellaFera et al* as applied to claims 10 and 33 above, and further in view of US. Patent Application No. 2002/0184308 to *Levy et al*.

Claims 5-7, 11-13, and 17-19 each depend from one of claims 10 or 33. Accordingly, Applicants submit that these dependent claims are allowable without the need for further comment from applicant.

Conclusion

Having addressed all issues set out in the office action, Applicants respectfully submit that the claims are in condition for allowance and respectfully request that the claims be allowed.

Respectfully submitted, and
S-signed pursuant to 37 CFR 1.4,

/Gero G. McClellan, Reg. No. 44,227/

Gero G. McClellan
Registration No. 44,227
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Attorney for Applicant(s)